Quality of Service for Wide Area Clusters

Colin Allison
Division of Computer Science,
University of St Andrews,
Scotland, KY16 9SS.
ca@dcs.st-andrews.ac.uk

Abstract

Quality of Service (QoS) is virtually synonymous with networked multimedia but could other types of distributed system benefit from QoS? This position paper examines the potential use of QoS in wide area cluster computing. The class of parallel applications that can benefit from clusters is currently restricted due to their relatively high latency and low throughput – consequences of conventional networking. LANs offer the best performance but also limit the scope for effective clustering to a single room or building. Reliability is another key difference between conventional multiprocessors and clusters: the former can reasonably be programmed with the error-free assumption but applications cannot be run on distributed clusters without programming against the potential for remote faults. Emergent wide area high speed switched networks have the potential to reduce latency, increase bandwidth, improve reliability and remove some of the geographical constraints from wide area clustering. Can QoS parameters be used or defined to exploit these capabilities?

1 Introduction

QoS is loosely defined in the ISO Open Distributed Processing program as follows: "The notion of QoS is a system or object property, and consists of a set of quality requirements on the collective behaviour of one or more objects" [1]. More particularly "the role of QoS management is to ensure that applications are able to specify and obtain the quality of service they require for their correct execution" [2]. In practice QoS must be supported by all components involved in providing the run time environment for a distributed application: host operating systems, network devices and network protocols. Operating systems which explicitly support QoS are relatively rare, for example [3]. Network protocols which support QoS include RSVP [4], IPv6 [5] and ATM. This paper focuses on ATM networks because they are currently the most advanced in QoS realities.[1] Table 1 summarises the current scope of approved ATM-based QoS parameters that may be negotiated on an end-to-end basis [5].

CBR	constant bit rate
rt-VBR	real time variable bit rate
nrt-VBR	non real time variable bit rate
UBR	unspecified bit rate
ABR	available bit rate
PCR	peak cell rate
SCR	sustainable cell rate
MBS	maximum burst size
MCR	minimum cell rate
CDV	cell delay variation
CDVT	cell delay variation tolerance
CDT	cell transfer delay
CLR	cell loss ratio

Service Categories

Attributes	CBR	rt-VBR	nrt- VBR	UBR	ABR
traffic parameters					
PCR, CDVT	√	√	√	√	√
SCR,MBS, CDVT	n/a	√	√	n/a	n/a
MCR	n/a	n/a	n/a	n/a	√
QoS parameters					
Peak-to-Peak CDV	√	√	-	-	-
Max CTD	√	√	-		
CLR	√	√	-	-	√
other					
Feedback	-	-	-	-	√

√ *attribute is specified for service,* - *not specified*

Table 1: Current Scope of ATM QoS.

QoS research to date has concentrated on the needs of real time continuous media and has only recently become an issue for other types of distributed applications. In [7] a Quality Description Language is described which allows "reliability requirements" to be expressed for distributed objects in CORBA environments. This paper looks at

[1] ATM is in no way mutually exclusive with IPv6 or RSVP. In fact there is a growing belief that they are well suited partners.

another area – *wide area cluster computing* – which could potentially benefit from QoS.

2 Wide Area Clustering

Although there are now numerous examples of installations where a group of desktop systems is purchased solely to act as a cluster [8,9,10] the holy grail of clustering is to tap the "free" resources in otherwise idle workstations. This quest is severely hampered by the *locality constraint.* The main architectural differences between clusters and integrated multiprocessors lie in their inter-node communications capabilities which can vary greatly in terms of bandwidth, latency and packet loss. The more spread out a cluster is the more it is prone to higher latency, lower throughput and arbitrary communications faults. It is not surprising that the most successful clustering strategies to date keep the distance between nodes to a minimum and connect them all to a single physical LAN segment or switch. In the more extreme cases, such as NOW [9], commodity components are packed into racks and connected by highly specialised network switches as part of a do-it-yourself supercomputer exercise.

However, the chances of dynamically constituting a suitable cluster from idle machines increase dramatically if a wider area containing a larger pool of possible participants is searched. Table 2 shows the type of criteria used in the query interface to the WARP (Wide Area Resource Program) resource discovery protocol (RDP) [11] and how the network scope of a query is constrained using IP Time–To-Live (TTL) values on the experimental IP multicast backbone.

Parameter	Example of values entered by cluster seeker
Hardware vendor	SGI, Sun, etc.
OS	unix 'uname -a', NT version 4.x, etc
Software	runtime libraries, compilers, Sun RPC, PVM, MPI, etc.,
Main memory, Mbytes	> 64 Mbyte
Virtual Memory, Mbytes	> 256 Mbyte
KDrhy/s	> 10
KWhet/s	> 10
Disk KB/s	> 4000
TCP KB/s	> 1000
Scope of Query, TTL	1 = LAN 15 = site e.g. campus 31 = high-bandwidth UK 63 = Europe 127 = high-bandwidth global

Table 2: The WARP resource discovery protocol

Note that the notion of QoS is already evident in the types of parameters specified in an RDP request, albeit in a far wider sense than that found in ATM. There is no problem in using RDP to include ATM QoS queries but at present there is no ATM query mechanism. An actual network admission attempt must therefore be made to test the water.

The concept of wide area clustering is (inevitably) being extended to the Web: Fox and Furmanski argue that clustering of future generation personal computers over the Web is the most promising technological path to obtaining peak performance in excess of 1 exaops[2] by the year 2007 [12]. Their calculations assume wide area high bandwidth connectivity – a scenario foreshadowed by the recent deployments of ATM between large groups of educational institutions separated by tens and hundreds of kilometers such as the SJIII network [13]. Commodity software that is currently shipped with ATM interface cards – *Lan Emulation* and *Logical IP Subnets* – allows workstations to be interconnected across wide area virtual networks bypassing conventional IP routers thereby relaxing the locality constraint from clustering. Unfortunately neither of these protocols allow for the strong QoS features of ATM to be exploited by parallel applications.

3 Discussion: QoS and reliability

A key difference between traditional multiprocessors and distributed systems is the (un)predictability of the environment. In a traditional multiprocessor environment code can reasonably be written with the "error-free" assumption i.e. a processor or interconnect fault is seen as a potentially rare occurrence that is the manufacturers or maintainers responsibility, it is not seen as a problem that should be addressed by the programmer. In a distributed system it is not always possible to make such assumptions. In the extreme case a cluster of IP nodes working on some shared task may be spread around the globe and subject to frequent uncertainties. Uncertainty stems from the impossibility of knowing whether an overdue communication is due to a slow network, a broken network, a crashed node or a slow node. Hence the distributed consensus problem, and the Fisher, Lynch and Paterson impossibility result [14]. Although protocols and algorithms which facilitate computational progress in the face of such uncertainty are still being researched, the proposed solutions tend to add costs in terms of latency and throughput and can be directly opposed to high performance. Can QoS be used to alleviate this problem?

QoS is an end-to-end system property and thus must be agreed by all network intermediaries and all intervening operating systems. If an application agrees a set of QoS parameters with the network and does not exceed it, and

[2] giga=10^9, tera=10^{12}, peta=10^{15}, exa=10^{18}

responds to flow control and other traffic feedback in the case of ABR (available bit rate), then, theoretically,

- there will be no packet loss
- worst case delay will be known
- quantified throughput will be known

Of course, in practice there are drawbacks: network overbooking and the small (10^{-9}) but not negligible bit error rate on fibre. Network overbooking is not transparent to end-points. If applied intelligently it should rarely interfere with agreed QoS parameters but it is still enough of a risk to necessitate, at the very least, an error detection protocol. There is however a significant difference between the low overhead of a lightweight error recovery protocol designed for a highly reliable communications infrastructure and the relatively high costs associated with providing reliable communications in conventional best effort networks. The "end-to-end" argument [15] is concerned with the balance between application level verification overhead and what can reasonably be expected from the underlying network. A great deal more can be expected from a QoS network than from conventional IP.

4 Cluster traffic patterns and QoS

There are two questions to consider: can clusters take advantage of the current QoS options, and, what new QoS options would be useful?

Table 3 is a brief exert from some traffic analysis work carried out by the NOW project [16] on closely-connected 16 and 32 processor clusters. It is included here simply as an example of cluster traffic characterisation from a well documented body of empirical data. (The purpose of their exercise was to validate the usefulness of parameters in their logP model). The average messages per processor per millisecond, average message interval in microseconds, and barrier interval in milliseconds characterises the synchronisation behaviour of a wide range of programs. The next two columns list the percentages of the total messages which were bulk messages and those which were read/request/reply. The remaining two columns show the average bandwidth measured in bytes at the communication layer in terms of bulk messages and small messages.

Could this type of traffic be supported in the wide area scenario? More information is needed to take advantage of current QoS parameters listed in Table 1. In particular, the maximum and minimum throughput and inter-message interval figures, a more refined analysis of message size distribution, and the actual time taken to execute a barrier (in addition to the interval). More information about message size would facilitate separation of control information and bulk data into separate virtual circuits with different and appropriate QoS. Barrier synchronisation for example needs virtually no bandwidth, and interestingly, Table 3 shows that barrier intervals on a closely connected system are relatively large for a wide range of applications – several being greater than 1 second. Such control signals do not require sustained or significant bandwidth per se, but low latency, maximum reliability, and multicast capability. This is a case where a new type of service would be beneficial for clustering.

In general we would not expect bulk data traffic to exhibit the regularity of continuous media. Burst sizes can easily exceed what the network is prepared to support although the *average* bandwidth utilised per node over the duration of a computation is well within its capabilities. Yet to benefit from QoS it is incumbent on the source end-point of an ATM circuit to conform to its agreement with the network. A parallel application must either understand its own traffic shape and negotiate suitable parameters to support it, or smooth and adapt its outgoing traffic to ensure it does not exceed agreed parameters. An application's "natural" shape could be estimated from recordings. Shaping and dynamic adaptation could usefully be done in a message passing layer, such as MPI [19].

Program	Ave. Msg./ Proc./ms	Ave. Msg. Int. (µs)	Ave. Barrier Int. (ms)	Percent Bulk Msg.	Percent Reads	Ave. Bulk Msg. KB/s	Ave. Small Msg. KB/s
Radix	164.76	6.1	408	0.01%	0	26.7	4,612.90
EM3D(write)	124.76	8	122	0	0	0.6	3,493.20
Barnes	18.94	52.8	279	23.25%	20.57%	110.4	407.1
P-Ray	6.4	156.2	1,120	47.85%	96.49%	358.5	93.5
Connect	5.45	183.5	47	0.06%	67.42%	0	152.5
Mur	4.7	212.6	11,778	49.99%	0	3,876.60	65.8
NOW-sort	1.22	817.4	1,834	49.82%	0	3,125.10	17.2
Radb	1.17	852.7	25	34.73%	0.04%	33.6	21.4

Table 3: Traffic Characterisation on a 32 node NOW cluster

NOW demonstrates the success of clustering but does not reflect any particular programming style. Bulk Synchronous Parallelism (BSP) [17] by contrast is a parallel programming methodology which is highly successful on shared memory multiprocessors but has failed to work satisfactorily in cluster mode using conventional LANs [18]. BSP's strength lies in its predictability and scalability for a given application and a known set of processor / memory pairs connected by a known bandwidth. Its performance problems in cluster mode are mainly due to the cost of distributing global synchronisations known as *supersteps*. It provides another example where a low latency multipoint timing signal would prove useful. The predictability of bandwidth afforded by network QoS would also help in predicting BSP performance on a cluster.

5 Conclusions

QoS is no longer solely a distributed multimedia issue. This paper has examined its potential in the rather different context of wide area cluster computing. Three interdependent aspects have been examined: locality, reliability and performance. Virtual networks in the form of Logical IP subnets and LAN emulation relax the locality constraints on clustering but neither allow access to the strong QoS features of ATM.

Two strategic options have been discussed: the exploitation of existing QoS parameters to reduce error–handling overheads by "pre-booking" network reliability, and the need for new classes of service relevant to cluster computing.

In order for a cluster application with relatively high traffic variance to take advantage of existing QoS facilities it should separate control from bulk data and be prepared to smooth its outputs. Preliminary tests on process-to-process communication across a 20Km ATM network traversing three switches have yet to record a lost or damaged message where the QoS parameters have conformed to the advertised policing ceiling. Barrier synchronisation execution times are lower than those obtained using the barrier instruction in a popular shared memory multiprocessor. Further tests are planned across an area of 130Km spanning seven switches.

A *multipoint simple event channel* is a new class of service which would prove useful for clusters. It requires low latency and maximum reliability but little bandwidth. The provision of new service classes and traffic shaping could usefully be incorporated into a message passing layer such as MPI which has already been adopted by parallel applications.

In summary, the aim of this work is to make wide area parallel computation more feasible through leveraging network QoS.

6 References

[1] ISO/IEC IS 10746-2.

[2] COULSON G, & de MEER J.B, Editor's Introduction, Distributed Systems Engineering Journal, Vol.4,No.1, March 1997, p1.

[3] LESLIE, I.M, et. al. *The Design and Implementation of an Operating System to Support Distributed Multimedia Applications*, in IEEE JSAC, 14,7 Sept.1996, pp 1280-1297.

[4] BRADEN, B, ZHANG, D, ESTRIN S, & JAMIN, S, *Resource ReSerVation Protocol (RSVP) -Version 1 Functional Specification*, Work in Progress Paper, IETF, January 1996.

[5] BRADNER S, and MANKIN A, *The Recommendation for the IP Next Generation Protocol*, RFC-1752, January 1995.

[6] The ATM Forum. Traffic Management Specification Version 4.0, April 1996.

[7] BAKKEN, D., SCHANTZ R, & ZINKY, J. *QoS Issues for wide-area CORBA-based object systems* in Proc. Second International Workshop on Object-Oriented, Real-Time Dependable Systems, IEEE, 1996.

[8] Dept. Aerospace Engineering cluster, Glasgow Univ. http://www.aero.gla.ac.uk/Research/CFD/Computing.

[9] CULLER D, et. al., Parallel Computing on the Berkeley Now, In JSPP'97 (9th Joint Symposium on Parallel Processing), Kobe, Japan. (http://now.cs.berkeley.edu/)

[10] RIDGE D, BECKER D, MERKEY P. *Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs* in Proc. IEEE Aerospace, 1997. (http://www.beowulf.org/papers/)

[11] ALLISON C, HARRINGTON P, HUANG F, & LIVESEY M.J, Scalable Services for Resource Management in Distributed and Networked Environments **in** SDNE'96, ed. P. Honeyman, IEEE Press, June 1996, pp98-105.

[12] FOX G.C, & FURMANSKI W, Petaops and Exaops: Supercomputing on the Web. IEEE Internet Computing, Vol.1,No.2, April 1997, pp 38-46.

[13] The SuperJANET III system http://www.ja.net/press_release/SJ_Brief.html

[14] FISCHER M.J, LYNCH N.A, & PATERSON, M.S, Impossibility of Distributed Consensus with One Faulty Process, *JACM*, **32**,2, (1985), 26-39.

[15] SALTZER, J.H, REED D.P., & CLARK D.D, End-To-End Arguments in System Design, ACM TOCS, Vo. 2, No. 3, August 1984. pp 277-288.

[16] MARTIN R.P, et. al, Effects of Communication Latency, Overhead, and Bandwidth in a Cluster Architecture, ICSA 24, Denver, June 1997. (http://now.cs.berkeley.edu).

[17] MCCOLL W.F, Bulk Synchronous Parallel Computing, **in,** Davy, J.R & Dew, P.M *Abstract Machine Models for Highly Parallel Computers* (Proc. 2nd Workshop on Abstract Machine Models, Leeds) OUP,1995. pp41-63.

[18] HILL, J.M.D & SKILLICORN, D.B, *Practical Barrier Synchronisation* in Proc. PDP'98, 6th Euromicro Workshop on Parallel and Distributed Programming, Madrid, Jan. 1998. pp438-444.

[19] MPI: The Message Passing Standard. http://www.mcs.anl.gov/mpi/